

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) A method for interrupt processing, comprising:
determining that an event has occurred;
determining a state of an event data structure, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of an Input/Output (I/O) device; [[and]]
writing an event entry into the event data structure in response to determining that the event has occurred and based on the state of the event data structure being one of armed or unarmed; and
issuing an interrupt in response to determining that the state of the event data structure is armed and that a condition exists to cause an interrupt.
2. (Currently Amended) The method of claim 1, ~~further comprising:~~
~~issuing an interrupt in response to determining that the state of the event data structure is armed and that a condition exists to cause an interrupt, wherein events are posted to the event data structure when the state of the event data structure is one of armed and unarmed and~~
wherein interrupts are not issued when the state of the event data structure is unarmed.
3. (Original) The method of claim 1, further comprising:
shutting down in response to determining that the state of the event data structure is undefined.
4. (Original) The method of claim 1, further comprising:
checking a structure state indicator to determine the state of the event data structure.
5. (Original) The method of claim 1, further comprising:
advancing a write indicator in the event data structure;

checking for an overflow condition; and
processing the overflow condition in response to determining that an overflow condition exists.

6. (Currently Amended) A method for interrupt processing, comprising:
determining that an interrupt has occurred;
reading an event entry in an event data structure in response to determining that the interrupt has occurred, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of an Input/Output (I/O) device;
[[and]]
updating a state of a structure state indicator to unarmed to indicate that interrupts are not allowed ;
determining whether an event code for the event entry that was read is clear; and
processing an event in response to determining that the event code is not clear.

7. (Currently Amended) ~~The method of claim 6, further comprising:~~ A method for interrupt processing, comprising:
determining that an interrupt has occurred;
reading an event entry in an event data structure in response to determining that the interrupt has occurred, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of an Input/Output (I/O) device;
and
updating [[the]] a state in [[the]] a structure state indicator to unarmed in response to determining that the interrupt has occurred.

8. (Currently Amended) The method of claim [[6]] 7 , further comprising:
determining whether to allow interrupts;
updating the state in a structure state indicator to armed in response to determining that interrupts are to be allowed; and

waiting for an interrupt.

9. (Original) The method of claim 6, further comprising:
determining whether a reset has occurred; and
updating the state in the structure state indicator to undefined in response to determining that the reset has occurred.

10. (Original) The method of claim 6, further comprising:
initializing the event data structure;
notifying an I/O device of the location of the event data structure; and
updating the state of the structure state indicator to unarmed.

11. (Cancelled)

12. (Currently Amended) The method of claim [[11]] 6, further comprising:
clearing the event code for the event entry; and
advancing a read indicator for the event data structure.

13. (Original) The method of claim 6, further comprising:
generating multiple event data structures for one I/O device, wherein events for different I/O device functions are associated with one of the multiple event data structures; and
dynamically mapping each of the multiple event data structures to a processor.

14. (Currently Amended) A system for [[in]] interrupt processing, comprising:
an Input/Output device coupled to a bus; and
circuitry at the Input/Output device operable to:
determine that an event has occurred;
determine a state of an event data structure, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of the Input/Output (I/O) device; [[and]]

write an event entry into the event data structure in response to determining that the event has occurred and based on the state of the event data structure being one of armed or unarmed; and

issue an interrupt in response to determining that the state of the event data structure is armed and that a condition exists to cause an interrupt.

15. (Currently Amended) The system of claim 14, wherein the circuitry is operable to:

~~issue an interrupt in response to determining that the state of the event data structure is armed and that a condition exists to cause an interrupt, wherein events are posted to the event data structure when the state of the event data structure is one of armed and unarmed and wherein interrupts are not issued when the state of the event data structure is unarmed.~~

16. (Original) The system of claim 14, wherein the circuitry is operable to:
shut down in response to determining that the state of the event data structure is undefined.

17. (Original) The system of claim 14, wherein the circuitry is operable to:
check a structure state indicator to determine the state of the event data structure.

18. (Original) The system of claim 14, wherein the circuitry is operable to:
advance a write indicator in the event data structure;
check for an overflow condition; and
process the overflow condition in response to determining that an overflow condition exists.

19. (Currently Amended) A system for [[in]] interrupt processing, comprising:
an Input/Output device driver coupled to a bus; and
circuitry at the Input/Output device driver operable to:
determine that an interrupt has occurred;

read an event entry in an event data structure in response to determining that the interrupt has occurred, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of an Input/Output (I/O) device; [[and]]

update a state of a structure state indicator to unarmed to indicate that interrupts are not allowed ;

determine whether an event code for the event entry that was read is clear; and
process an event in response to determining that the event code is not clear.

20. (Currently Amended) ~~The system of claim 19, wherein the circuitry is operable to:—A system for interrupt processing, comprising:~~

an Input/Output device driver coupled to a bus; and
circuitry at the Input/Output device driver operable to:
determine that an interrupt has occurred;
read an event entry in an event data structure in response to determining that the interrupt has occurred, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of an Input/Output (I/O) device;
and

update [[the]] a state in [[the]] a structure state indicator to unarmed in response to determining that the interrupt has occurred.

21. (Currently Amended) The system of claim [[19]] 20 , wherein the circuitry is operable to:

determine whether to allow interrupts;
update the state in a structure state indicator to armed in response to determining that interrupts are to be allowed; and
wait for an interrupt.

22. (Original) The system of claim 19, wherein the circuitry is operable to:

determine whether a reset has occurred; and
update the state in the structure state indicator to undefined in response to determining that the reset has occurred.

23. (Original) The system of claim 19, wherein the circuitry is operable to:
initialize the event data structure;
notify an I/O device of the location of the event data structure; and
update the state of the structure state indicator to unarmed.

24. (Cancelled)

25. (Currently Amended) The system of claim [[24]] 19 , wherein the circuitry is operable to:

clear the event code for the event entry; and
advance a read indicator for the event data structure.

26. (Original) The system of claim 19, wherein the circuitry is operable to:
generate multiple event data structures for one I/O device, wherein events for different I/O device functions are associated with one of the multiple event data structures; and
dynamically map each of the multiple event data structures to a processor.

27. (Currently Amended) An article of manufacture for interrupt processing, wherein the article of manufacture at an Input/Output device is operable to:

determine that an event has occurred;
determine a state of an event data structure, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of the Input/Output (I/O) device; [[and]]

write an event entry into the event data structure in response to determining that the event has occurred being one of armed or unarmed; and

issuing an interrupt in response to determining that the state of the event data structure is armed and that a condition exists to cause an interrupt.

28. (Currently Amended) The article of manufacture of claim 27, ~~wherein the article of manufacture is operable to:~~

~~issue an interrupt in response to determining that the state of the event data structure is armed and that a condition exists to cause an interrupt, wherein events are posted to the event data structure when the state of the event data structure is one of armed and unarmed and wherein interrupts are not issued when the state of the event data structure is unarmed.~~

29. (Original) The article of manufacture of claim 27, wherein the article of manufacture is operable to:

shut down in response to determining that the state of the event data structure is undefined.

30. (Original) The article of manufacture of claim 27, wherein the article of manufacture is operable to:

check a structure state indicator to determine the state of the event data structure.

31. (Original) The article of manufacture of claim 27, wherein the article of manufacture is operable to:

advance a write indicator in the event data structure;

check for an overflow condition; and

process the overflow condition in response to determining that an overflow condition exists.

32. (Currently Amended) An article of manufacture for interrupt processing, wherein the article of manufacture at an Input/Output device driver is operable to:

determine that an interrupt has occurred;

read an event entry in an event data structure in response to determining that the interrupt has occurred, wherein the event data structure includes one or more entries, and wherein each of

the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of an Input/Output (I/O) device; [[and]]

update a state of a structure state indicator to unarmed to indicate that interrupts are not allowed ;

determine whether an event code for the event entry that was read is clear; and
process an event in response to determining that the event code is clear.

33. (Currently Amended) ~~The article of manufacture of claim 32, wherein the article of manufacture is operable to:~~ An article of manufacture for interrupt processing, wherein the article of manufacture at an Input/Output device driver is operable to:

determine that an interrupt has occurred;

read an event entry in an event data structure in response to determining that the interrupt has occurred, wherein the event data structure includes one or more entries, and wherein each of the entries is capable of storing event specific parameters and an event code field that identifies at least one of an event source and function of an Input/Output (I/O) device; and

update [[the]] a state in [[the]] a structure state indicator to unarmed in response to determining that the interrupt has occurred.

34. (Currently Amended) The article of manufacture of claim [[32]] 33 , wherein the article of manufacture is operable to:

determine whether to allow interrupts;

update the state in a structure state indicator to armed in response to determining that interrupts are to be allowed; and

wait for an interrupt.

35. (Original) The article of manufacture of claim 32, wherein the article of manufacture is operable to:

determine whether a reset has occurred; and

update the state in the structure state indicator to undefined in response to determining that the reset has occurred.

36. (Currently Amended) The article of manufacture of claim 32, wherein the article of manufacture is operable to:

- initialize the event data structure;
- notify[[,]] an I/O device of the location of the event data structure; and
- update the state of the structure state indicator to unarmed.

37. (Cancelled)

38. (Currently Amended) The article of manufacture of claim [[37]] 32 , wherein the article of manufacture is operable to:

- clear the event code for the event entry; and
- advance a read indicator for the event data structure.

39. (Original) The article of manufacture of claim 32, wherein the article of manufacture is operable to:

- generate multiple event data structures for one I/O device, wherein events for different I/O device functions are associated with one of the multiple event data structures; and
- dynamically map each of the multiple event data structures to a processor.